

ИНФОРМАЦИОННАЯ СИСТЕМА «ЕХОН.ЦИМ»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ,
ПРЕДОСТАВЛЕННОГО ДЛЯ ПРОВЕДЕНИЯ ЭКСПЕРТНОЙ ПРОВЕРКИ

Содержание

Термины и определения	3
1. Общие сведения	4
1.1. Назначение	4
1.2. Автоматизируемые функции	4
2. Установка Системы	5
2.1. Системные требования	5
2.2. Начальная конфигурация	5
2.3. Установка окружения	5
2.4. Установка зависимостей	5
2.5. Запуск Системы	6

Термины и определения

В настоящем документе используются следующие термины:

Термин	Определение
Дамп	Содержимое рабочей памяти одного процесса, ядра или всей операционной системы. Также может включать дополнительную информацию о состоянии программы или системы, например значения регистров процессора и содержимое стека
Доменное имя	Символьное имя, служащее для идентификации областей, которые являются единицами административной автономии в сети Интернет, в составе вышестоящей по иерархии такой области. Каждая из таких областей называется доменом
Система	Информационная система «Ехон.ЦИМ»

В настоящем документе используются следующие сокращения:

Сокращение	Определение
ЦИМ	Цифровая информационная модель
ОС	Операционная система
ПО	Программное обеспечение
DNS	Domain name server — приложение, предназначенное для ответов на DNS-запросы по соответствующему протоколу
HDD	Hard (magnetic) disk drive — запоминающее устройство (устройство хранения информации, накопитель) произвольного доступа, основанное на принципе магнитной записи
IP	Internet Protocol — маршрутизируемый протокол сетевого уровня стека TCP/IP
RAM	Random Access Memory — один из видов памяти компьютера, позволяющий одновременно получить доступ к любой ячейке (всегда за одно и то же время, вне зависимости от расположения) по ее адресу на чтение или запись

1. Общие сведения

1.1. Назначение

Ехон.ЦИМ — цифровая платформа, предназначенная для формирования требований к ЦИМ, для загрузки, просмотра, согласования, публикации и хранения ЦИМ, отражающих проектные решения по разным разделам, а также проверки ЦИМ, в том числе на соответствии требованиям (далее — Система).

Система предоставляет единое пространство, позволяющее объединить всех участников процессов согласования и проверки ЦИМ:

- Заказчик;
- Технический заказчик;
- Генеральный проектировщик;
- Проектировщик;
- Генеральный подрядчик;
- Подрядчик;
- Строительный контроль;

1.2. Автоматизируемые функции

Система позволяет автоматизировать выполнение следующих функций:

- Формирование требований для технического задания к ЦИМ, контроль их версионности;
- Хранение и контроль версионности ЦИМ с привязкой к документации ПД, РД;
- Загрузка и отображение ЦИМ в формате IFC;
- Работа с 3D-представлением ЦИМ (навигация, управление отображением, проведение измерений, создание секущих плоскостей и разрезов);
- Отображение иерархической структуры IFC и дерева элементов, фильтрация элементов ЦИМ по атрибутам;
- Отображение свойства выделенного элемента ЦИМ;
- Добавление замечаний к элементам ЦИМ для дальнейшего отображения в документах, формируемых в модулях;
- выгрузка отчетов в формат .xlsx;
- Проверка атрибутивного состава и значений атрибутов ЦИМ на соответствие сформированным ранее требованиям;

2. Установка Системы

2.1. Системные требования

Для установки требуется ОС Linux, kernel 4.15 или выше. Рекомендуется использовать дистрибутив Debian 10 или выше.

Для минимальной установки необходимо 16 ГБ RAM и 50 ГБ HDD. Рекомендуется 32 ГБ RAM, объем диска зависит от количества загружаемых в Систему пользователями файлов.

Для запуска необходимо установить ПО `docker` и `docker-compose` Рекомендуется также установка: **`curl, git, iotop, less, mlocate, tcpdump, telnet, traceroute, vim, make`**.

Для получения пакетов Системы из репозитория понадобится доступ в Интернет.

Данная инструкция описывает установку Системы для целей разработки или тестирования. При установке в продуктивной среде необходимо предпринять меры по защите и предотвращению доступа к базам данных и другой инфраструктуре проекта.

2.2. Начальная конфигурация

Для работы Системы необходим DNS-сервер и доменное имя.

Для примера в данной инструкции будет использоваться домен первого уровня `exon`. Если необходима работа с другим именем домена, понадобятся дополнительные настройки.

1) В файле `.env` необходимо установить значение переменной `HOST "exon"` (для работы локально на одном компьютере, либо любое другое значение вашего домена):
`HOST=exon` .

Для корректной работы необходимо, чтобы в DNS существовала wildcard запись (`*.exon` либо отдельные записи для поддоменов: **`pdftron, traefik, rmq, kibana`**) для доменов нижнего уровня на тот же IP-адрес. Если нет возможности использовать DNS-сервер, можно добавить в файл `/etc/hosts` строку: `127.0.0.1 exon pdftron.exon kibana.exon traefik.exon rmq.exon`

2) В файле `config/application.yml` заменить домен `exon` на актуальный.

3) В файле `config/frontSettings.json` заменить домен `exon` на актуальный в переменных: `"api": "http://exon/api", "pdfTronServer": "http://pdftron.exon", "urlIdentity": "http://exon/auth", "urlReactApp": "http://exon"`.

2.3. Установка окружения

Для работы Системы необходимы базы данных: `Mongodb` и `Postgresql`. Также необходимы сервисы:

- `Rabbitmq`;
- `Redis`;
- `Elasticsearch`;
- `Kibana`;
- `Traefik`.

Конфигурация данных сервисов описывается в файле `infrastructure.yml` для `docker-compose`. Запуск всех необходимых сервисов выполняется командой: `sh docker-compose -f infrastructure.yml up -d`

После запуска сервисов необходимо сконфигурировать доступ к ним согласно инструкциям на сайтах разработчиков. Конфигурирование баз данных выходит за рамки данного документа.

2.4. Установка зависимостей

Для корректной работы Системы необходимы сервисы:

- Keycloak (аутентификация и авторизация пользователей);
- pdftron (отображение и редактирование pdf документов через браузеры);
- config (сервер для хранения настроек приложения).

Перед запуском сервисов необходимо создать базу для keycloak и импортировать в нее дамп **keycloak.sql** с начальными настройками keycloak для проекта.

Конфигурация данных сервисов описывается в файле **dependencies.yml** для docker-compose. Запуск всех необходимых сервисов выполняется командой: `sh docker-compose -f dependencies.yml up -d`

Если доменное имя отличается от **exon**, после запуска необходимо прописать настройки клиента аутентификации. Для этого:

- 1) С помощью любого веб-браузера перейти по адресу <http://exon/auth> (заменить **exon** на используемое имя домена). Имя и пароль по умолчанию: **admin/admin**.
- 2) Выбрать **realm SpringBoot**, перейти в раздел **Clients**, для клиента ExonReactApp прописать корректные url в параметры **Root URL, Valid Redirect URIs, Admin URL, Web Origins**.
- 3) Указать пароль для пользователя доступа к API keycloak.
- 4) Выбрать **realm Master**, перейти в раздел **Users**, найти пользователя **admin**, перейти на вкладку **Credentials** и установить пароль.
- 5) Перед сохранением убрать флаг **temporary**. Данный пароль вписать в конфигурационный файл **config/user_service.yml** в секцию **keycloak-config.admin-password**.

2.5. Запуск Системы

Конфигурация сервисов описывается в файле **services.yml** для docker-compose. Запуск всех необходимых сервисов выполняется командой: `sh docker-compose -f services.yml up -d`